Computer Systems, Inc.
MERCURY

Portable Waveforms for Heterogeneous Processing Platforms

Jim Kulp, Murat Bicer
Mercury Computer Systems, Inc.

JPO Firmware Portability Workshop
29-30 April 2004

The Ultimate Performance Machine

Computer Systems, Inc.
MERCURY
The Ultimate Performance Machine

SCA Waveforms/Applications

SCA waveforms consist of functionality implemented as:

- **Software components written in C++/POSIX/CORBA and communicating via a CORBA ORB (*Soft components*)**
- **Program loads for programmable devices that can load software/firmware not written as software components (*Firm components*)**
- **Fixed function devices including hard ASIC and I/O (*Hard components*)**

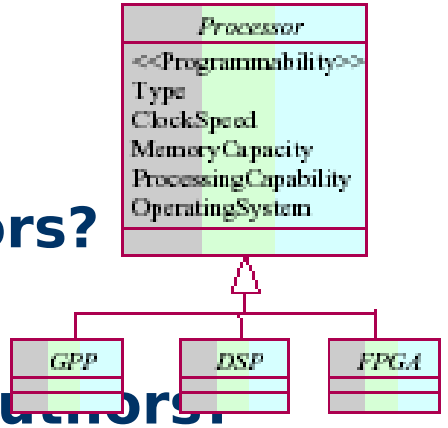
Computer Systems, Inc.
MERCURY
The Ultimate Performance Machine

Where is the Portability?

- For *Soft* components
  - SCA specification provides portability
- For *Firm* components
  - No portability due to lack of standardization
    - How they are written (C++ for soft)
    - How they talk to their environment (POSIX for soft)
    - How their interactions are defined (IDL for soft)
    - How they talk to each other (CORBA for soft)
  - No simple plug&play with *soft* components
    - No similar & compatible interface definition
    - Can't just be another implementation
    - No direct interoperation with *soft* components
- For *Hard* components
  - No portability, but some API standardization
  - "Logical device" is soft proxy for usage of device as hard component

Reconsidering *firm* components

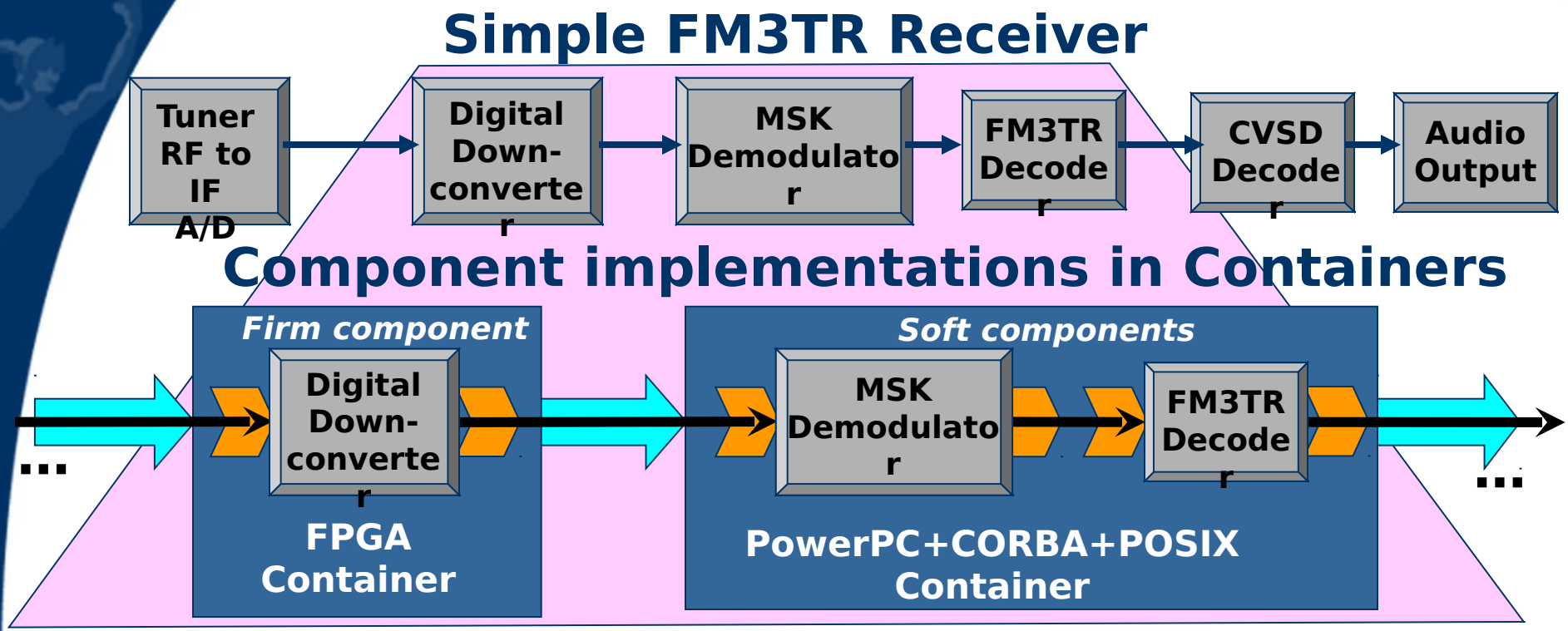
Let's challenge every difference from soft



- **Why can't *firm* components:**
  - **Have Software Component Descriptors?**
  - **Use Interactions defined in IDL?**
  - **Generate local APIs for component authors?**
  - **Interact transparently and portably with *soft* components?**
  - **Be used for any processing function (not just "signal processing front end")?**

They just use a "processor" with a different language and a different IDL mapping

Components and Containers



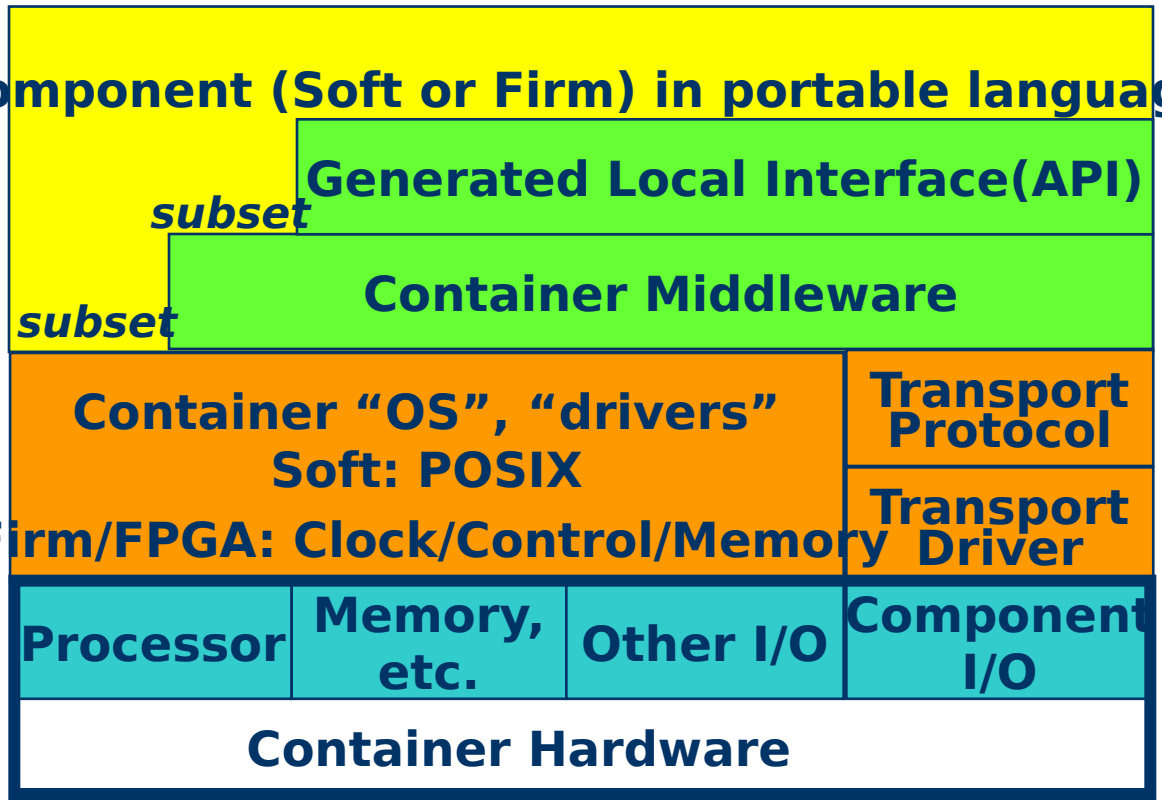
Components talk to their containers.
Important interface for portability of components. APIs used by component authors.

Containers talk to containers.
Important interface for interoperability/plug&play of containers (e.g. boards)

Communication between components/containers conveyed by their Protocols/networks/buss

Anatomy of a Container

A container is a place to run components. Exactly as an SCA Executable Device runs soft components written in C++/CORBA/POSIX.



Soft Container (per SCA)	Firm/FPGA Container (examples)
C++/Ada/ Java etc.	VHDL/IEEE 1076-2002?
CORBA Stub/Skel	IDL→OCP →VHDL?
CORBA ORB	Messaging IP Core?
e.g. TCP/IP	e.g. RapidIO or SerialLite?
e.g. NIC/Bus Driver	e.g. XAUI
e.g. PCI, Ethernet	e.g. LVDS